

ROBOFUN.RO
LECȚIA XII

CURS GRATUIT

ARDUINO ȘI ROBOTICĂ

Conexiune WIFI

Proiect :

**Stocarea online a datelor culese
de Arduino, folosind Google Docs**

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs
CC BY-NC-ND



Codul sursa din acest document este licentiat

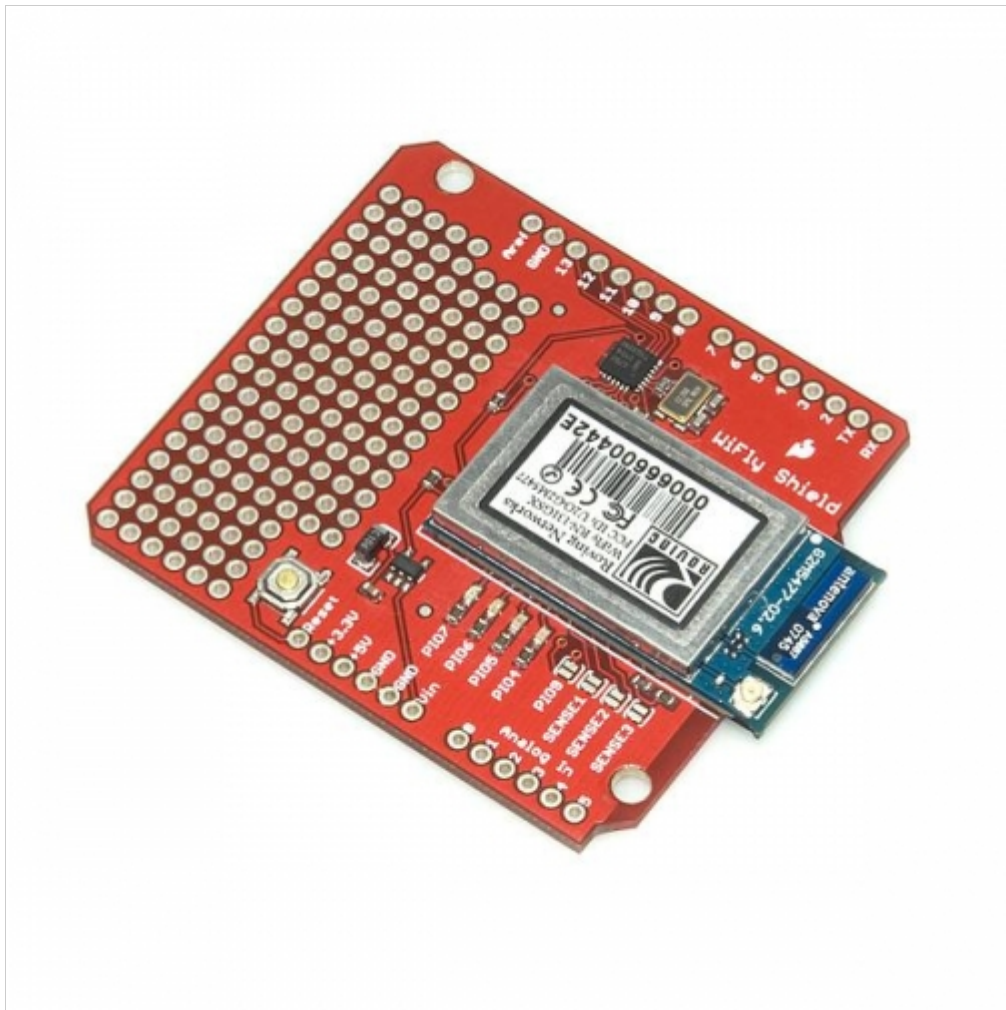
Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

WiFly Shield

Daca ai nevoie sa obtii informatii direct din Internet folosind Arduino, sau sa ai un server web rulant pe Arduino si vrei sa-l accesezi tu din Internet, si toate astea fara fir de retea, atunci WiFly Shield este ceea ce ai nevoie. Functioneaza prin WIFI, se conecteaza la un router si iti ofera conexiune la Internet pe Arduino. Daca nu iti este foarte clar cum functioneaza o retea de calculatoare (termeni gen IP, DNS, MAC, DHCP iti suna ciudat), atunci iti recomand sa citesti mai intai sectiunea in care se discuta despre shield-ul Ethernet pentru Arduino, sectiune in care am prezentat si aceste concepte.

Libraria de care vei avea nevoie in codul de mai jos o gasesti in aceasta pagina - http://www.robofun.ro/wifly_shield, descarc-o si instaleaz-o in mediul tau Arduino inainte de a rula exemplul de mai jos.



WiFly Shield – Client Web

```
//pune "" la ambele daca nu ai parola la WIFI
#define SSID "networkID"
#define PASSPHRASE "password"

#include "WiFly.h"
#include "Credentials.h"

Client client("google.com", 80);

void setup() {
  Serial.begin(9600);
  WiFly.begin();
  if (!WiFly.join(SSID, PASSPHRASE)) {
    Serial.println("Conectare la retea WIFI esuata.");
  }

  Serial.println("Conectare in progres...");

  if (client.connect()) {
    Serial.println("Conectare reusita !");
    client.println("GET /search?q=arduino HTTP/1.0");
  }
}
```



```
        client.println();
    } else {
        Serial.println("Conectare esuata");
    }
}

void loop() {
    if (client.available()) {
        char c = client.read();
        Serial.print(c);
    }

    if (!client.connected()) {
        Serial.println();
        Serial.println("disconnecting.");
        client.stop();
        for(;;)
            ;
    }
}
```

Daca ai parcurs deja sectiunea despre Ethernet shield, vei vedea ca ceea ce avem mai sus seamana foarte mult cu codul de acolo. Practic, se modifica doar modul de conectare la retea. Sunt de remarcat cele doua constante de la inceputul programului, care iti permit sa declari care este identificatorul tau de retea WIFI si care este parola. Daca ai o retea fara parola, pune "" in loc de parola si de id de retea.

Codul se conecteaza la serverul google.com si afiseaza in Serial Monitor rezultatele cautarii pentru termenul "arduino" (tu vei vedea informatie in format HTML, asa cum am explicat la sectiunea despre Ethernet Shield).

WiFly Shield – Server Web

```
//pune "" la ambele daca nu ai parola la WIFI
#define SSID "networkID"
#define PASSPHRASE "password"

#include "WiFly.h"
#include "Credentials.h"

Server server(80);

void setup() {
    WiFly.begin();
    if (!WiFly.join(SSID, PASSPHRASE)) {
        Serial.println("Conectare la retea WIFI esuata.");
    }

    Serial.begin(9600);
    Serial.print("IP: ");
```

```
Serial.println(WiFly.ip());

server.begin();
}

void loop() {
  Client client = server.available();
  if (client) {
    boolean current_line_is_blank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        if (c == '\n' && current_line_is_blank) {
          // send a standard http response header
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println();

          for (int i = 0; i < 6; i++) {
            client.print("* pe portul analogic ");
            client.print(i);
            client.print(" s-a citit valoarea ");
            client.print(analogRead(i));
            client.println("<br />");
          }
          break;
        }
        if (c == '\n') {
          current_line_is_blank = true;
        } else if (c != '\r') {
          current_line_is_blank = false;
        }
      }
    }
    delay(100);
    client.stop();
  }
}
```

Exemplul de mai sus creaza un server web care ruleaza pe placa Arduino si la fiecare cerere din browser raspunde cu valorile de pe porturile analogice, ca mai jos. IP-ul server-ului este alocat in mod dinamic de router, asa ca, pentru a sti care este adresa pe care o folosesti in browser, iti recomand sa deschizi Serial Monitor si sa vezi in debug IP-ul alocat placii.

Stocarea online a datelor culese de Arduino, folosind Google Docs

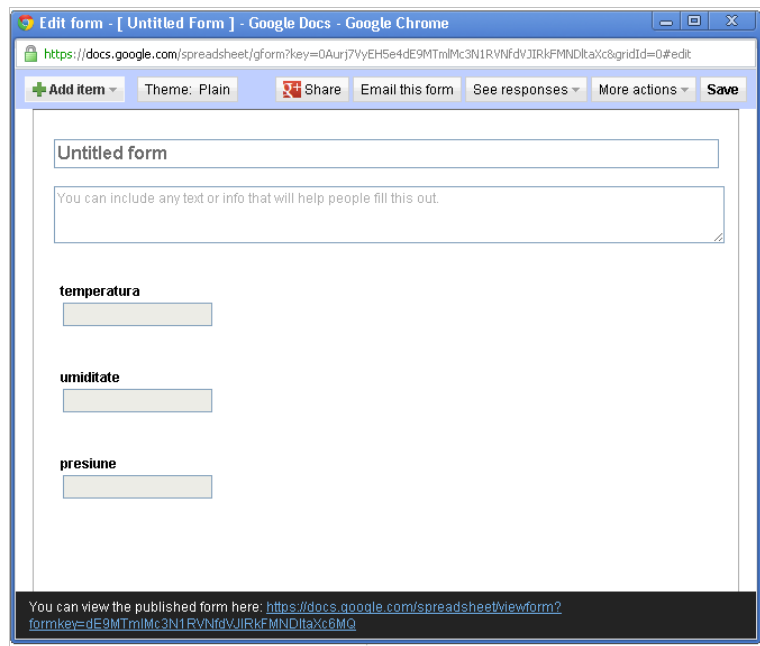
Google Docs reprezinta o solutie excelenta pentru a loga online date culese cu Arduino. Temperatura, umiditatea sau presiunea atmosferica sunt exemple foarte bune in acest sens. Trimiterea lor online intr-un document Excel se bazeaza pe functionalitatea Forms, oferita de Google Docs. Ideea de

principiu este de a genera un request HTTP din Arduino, care request sa fie perfect similar cu cel generat de pagina din form-ul Google. Astfel, Google este "pacalit" sa creada ca datele trimise de Arduino sunt de fapt trimise de un form creat cu Google Docs.

Pentru acest proiect ai nevoie un Arduino cu conectare la Internet. Ai de ales intre Arduino Ethernet sau Arduino UNO + Ethernet Shield. Senzorii sunt la alegerea ta, in functie de ce date vrei sa trimiti in Google Docs. Pentru exemplul de mai jos, eu am ales un BMP085 (ca sa masor presiunea atmosferica si temperatura), si un HIH-4030 pentru masurarea umiditatii. Evident ca tu poti alege ce senzori doresti.

Mai departe, acceseaza <http://docs.google.com> si creaza un document tip spreadsheet ("CREATE", apoi "Spreadsheet"). Ar trebui sa vezi un document similar unui document Excel. Din meniul "Tools", alege "Create a form". Form-ul creat ar trebui sa aiba cate un camp de tip "Text" pentru fiecare senzor pe care il vei trimite catre Google Docs. Spre exemplu, eu am creat un camp "temperatura", un camp "umiditate" si un camp "presiune", toate de tip "Text".

Selecteaza apoi din meniul "Form (0)" intrarea "Go to live form". Vei vedea formularul creat de Google Docs pentru culegerea datelor. Mai departe, va trebui sa analizezi codul sursa al paginii (in functie de browser-ul pe care il folosesti, acest lucru se face diferit - spre exemplu in Chrome trebuie sa dai click dreapta si apoi sa selectezi "View page source"). In codul sursa al paginii localizeaza sirul de caractere "formkey=". Vei gasi ceva de genul "*formkey=dE9MTmIMc3N1RVNfdVJRkFMNDItaXc6MQ*". Acest sir de caractere reprezinta identificatorul unic al formularului tau, identificator pe care Arduino il va utiliza ca sa trimita informatiile. Localizeaza si campurile in care introduci informatia utila (va fi simplu, pentru ca vor fi prefixate de numele campurilor pe care le-ai dat tu cand ai creat formularul - vezi si imaginea de mai jos). Numele acestor campuri ar trebui sa fie ceva de genul "*entry.0.single*", "*entry.1.single*", "*entry.2.single*" si tot asa pentru toate campurile care le-ai declarat.



The screenshot shows a web browser window titled "Edit form - [Untitled Form] - Google Docs - Google Chrome". The address bar displays a long URL starting with "https://docs.google.com/spreadsheet/gform?". Below the address bar is a toolbar with buttons: "Add item", "Theme: Plain", "Share", "Email this form", "See responses", "More actions", and "Save". The main content area contains a form titled "Untitled form" with a text box for a description: "You can include any text or info that will help people fill this out." Below this are three input fields, each with a label to its left: "temperatura", "umiditate", and "presiune". At the bottom of the form, a black bar contains the text: "You can view the published form here: [https://docs.google.com/spreadsheet/viewform?formkey=dE9MTmIMc3N1RVNfdVJIRkFMNDItaXc6MQ](\"https://docs.google.com/spreadsheet/viewform?formkey=dE9MTmIMc3N1RVNfdVJIRkFMNDItaXc6MQ\")".

Mai departe, ca sa trimita datele catre Google, Arduino va trimite o cerere HTTP catre serverul Google, care cerere va fi identica cu cea pe care o trimite browser-ul atunci cand apesi pe butonul "Submit" din formular. Practic, serverul Google va fi convins ca acele date vin din formularul obisnuit si nu din Arduino.


```
ype="http://schema.org/CreativeWork/FormObject"><meta itemprop="name" content="Untitled form">
google.com/spreadsheet/formthumb?formkey=dE9WTmlMc3NlRVNfdVJIRkFMNDltaxC6MQ">
le.com/spreadsheet/embeddedform?formkey=dE9WTmlMc3NlRVNfdVJIRkFMNDltaxC6MQ">
om/docs/spreadsheets/forms/favicon_32x32.png"
e">Untitled form</h1>

le.com/spreadsheet/formResponse?formkey=dE9WTmlMc3NlRVNfdVJIRkFMNDltaxC6MQ&amp;ifq" method="POST"
```

```
ak" style="display:none;">
orm action="https://docs.google.com/spreadsheets/formResponse?formkey=dG9HWmNXWjNRdWhBWG5ITlpNeUVBU2c6MQ;"
od">
-text"><div class="ss-form-entry"><label class="ss-q-title" for="entry_0">temperatura
" for="entry_0"></label>
="entry.0.single" value="" class="ss-q-short" id="entry_0"></div></div></div>
ox-good">
-text"><div class="ss-form-entry"><label class="ss-q-title" for="entry_1">umiditate
" for="entry_1"></label>
="entry.1.single" value="" class="ss-q-short" id="entry_1"></div></div></div>
ox-good">
-text"><div class="ss-form-entry"><label class="ss-q-title" for="entry_2">presiune
" for="entry_2"></label>
="entry.2.single" value="" class="ss-q-short" id="entry_2"></div></div></div>
me="pageNumber" value="0">
me="backupCache" value="">
```

Codul sursa integral pentru Arduino il gasesti aici - <http://robofun.ro/blog/cum-sa-trimiti-loghezi-pe-google-docs-temperatura-presiunea-atmosferica-si-umiditatea-direct-din-arduino> . In cele ce urmeaza vom analiza partile interesante din cod. Chiar la inceput ai o declaratie de forma `"char formkey[] = "dG9HWmNXWjNRdWhBWG5ITlpNeUVBU2c6MQ";"` Aici va trebui sa modifichi cheia din cod cu cheia din documentul tau, pe care ai obtinut-o mai sus. Mai departe, in functia `"loop"` vei gasi denumirile campurilor din document, pe care le-ai identificat deja mai sus. Va trebui sa modifichi aceasta zona din cod pentru a trimite exact informatiile culese de tine. Spre exemplu, in cazul meu codul este ca mai jos :

```
String data;
data+="";
```

```
data+="entry.2.single=";  
data+=temperatura;  
data+="&entry.3.single=";  
data+=presiune;  
data+="&entry.4.single=";  
data+=umiditate;  
data+="&submit=Submit";
```

Adaug in varianta *data* rand pe rand denumirea unui camp si valoarea acestuia, citita de Arduino. Mai departe, codul trimite request-ul catre serverul Google pentru a inregistra informatia in document.

Aceasta a fost lectia 12. In final, as vrea sa te rog sa ne oferi feedback asupra acestei lectii, pentru a ne permite sa le facem mai bune pe urmatoarele.

Este vorba despre un sondaj cu 4 intrebari (oricare este optionala), pe care il poti accesa [dand click aici](#).

Sau ne poti contacta direct prin email la contact@robofun.ro .

Iti multumim,

Echipa [Robofun.RO](#)